

Đề thi CẤU TRÚC DỮ LIỆU VÀ GIẢI THUẬT

Học kỳ I, 2016-2017

Thời gian: 100 phút

Lưu ý: Đề thi có 02 trang. Sinh viên không được sử dụng tài liệu.

Câu 1 (1 điểm)

a. Sắp xếp độ phức tạp tính toán (ký hiệu O) theo thứ tự tăng dần:

$O(n)$, $O(1)$, $O(2^n)$, $O(n \cdot \log_2 n)$, $O(n^2)$, $O(\log_2 n)$, $O(n^3)$.

b. Tìm độ phức tạp tính toán cho hàm `count()` dưới đây:

```
int count(int n) {
    int result = 0;
    while (n != 0) {
        result += 1;
        n /= 2;
    }
    return result;
}
```

Câu 2 (1 điểm)

Giả sử có ngăn xếp (stack) S gồm các giá trị $[1, 2, 3]$ (trong đó 3 ở đầu ngăn xếp (bên phải)); và có một hàng đợi (queue) Q rỗng (với đầu hàng đợi ở bên phải, và đuôi ở bên trái).

Chỉ sử dụng S , Q và thêm một biên duy nhất nếu cần (không được sử dụng thêm bất kỳ cấu trúc dữ liệu nào khác), hãy liệt kê các thao tác để có được ngăn xếp S như sau:

a. $[1, 3]$

b. $[3, 2, 1]$

Câu 3 (1,5 điểm)

Theo ký hiệu O , hãy cho biết:

a. Thời gian tìm kiếm trung bình một phần tử trong mảng (array) của n phần tử đã được sắp xếp là bao nhiêu ?

b. Thời gian tìm kiếm trung bình một phần tử trong danh sách liên kết đơn (single linked-list) của n phần tử đã được sắp xếp là bao nhiêu ?

c. Thời gian tìm kiếm trung bình một phần tử trong bảng băm (hash table) có n phần tử là bao nhiêu ?

d. Thời gian tìm kiếm xấu nhất một phần tử trong cây tìm kiếm nhị phân (binary search tree) có n đỉnh là bao nhiêu ?

e. Thời gian tìm kiếm xấu nhất một phần tử trong cây cân bằng AVL có n đỉnh là bao nhiêu ?

g. Thời gian tìm kiếm xấu nhất một phần tử trong cây thứ tự bộ phận (heap) có n đỉnh là bao nhiêu ?

Câu 4 (1,5 điểm)

a. Hãy sắp xếp mảng các số $\{1, 4, 10, 9, 6, 5, 3, 7, 2, 8\}$ theo thứ tự tăng dần bằng thuật toán Sắp xếp hòa trộn (Merge sort).

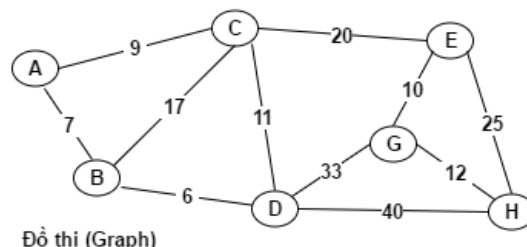
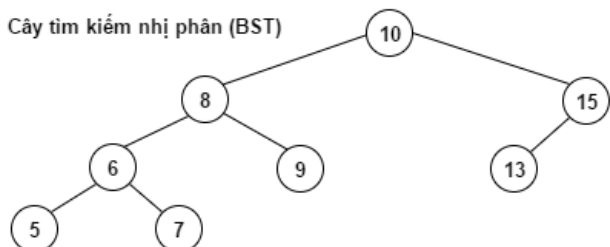
b. Cho biết độ phức tạp tính toán trung bình và xấu nhất của Sắp xếp hòa trộn ?

c. Thế nào là thuật toán sắp xếp ổn định (stable sort) ? Sắp xếp hòa trộn có phải là thuật toán sắp xếp ổn định không ?

Câu 5 (2 điểm)

Cho cây tìm kiếm nhị phân (binary search tree - BST) T như trong hình vẽ (ở dưới, bên trái).

- Cho biết thứ tự các đỉnh khi duyệt cây T theo thứ tự trong (inorder) và theo thứ tự trước (preorder).
- Trình bày các bước để thêm đỉnh có khoá 14 vào cây T .
- Cây T ban đầu (trước khi thêm đỉnh) có phải là cây cân bằng AVL không? Cây mới (sau khi thêm vào đỉnh khoá 14) có phải là cây cân bằng AVL không? Vì sao?
- Trình bày các bước để xóa đỉnh có khoá 10 ra khỏi cây T ban đầu.



Câu 6 (1 điểm)

Cho đồ thị (graph) vô hướng có trọng số như trong hình vẽ (ở trên, bên phải). Sử dụng một trong hai thuật toán Prim hoặc Kruskal, hãy tìm cây bao trùm nhỏ nhất (minimum spanning tree - MST) cho đồ thị.

Câu 7 (1 điểm)

Cho dãy n số nguyên khác nhau đôi một. Hãy trình bày thuật toán đếm số cặp hai số có tổng là 0 (không). Đánh giá độ phức tạp của thuật toán?
 Ví dụ: -20, -10, -9, 1, 4, 9, 11, 20. Kết quả trả về: 2 (do có hai cặp [-20, 20] và [-9, 9]).

Câu 8 (1 điểm)

Cho một lưới 2 chiều có n hàng và m cột. Mỗi ô của lưới có giá trị là 0 hoặc 1. Hai ô được gọi là kết nối với nhau nếu chúng cùng là 1, và nằm sát nhau trên cùng hàng hoặc cùng cột (tức ô $[i, j]$ có thể kết nối với 4 ô $[i-1, j]$, $[i+1, j]$, $[i, j-1]$, $[i, j+1]$). Đường đi giữa 2 ô $[r1, c1]$ và $[r2, c2]$ là dãy các ô kết nối nhau bắt đầu từ $[r1, c1]$ và kết thúc tại $[r2, c2]$. Như thế, giữa 2 ô có thể có không, một hay nhiều đường đi. Hãy trình bày ngắn gọn thuật toán (ý tưởng, các bước chính, cấu trúc dữ liệu/biến cần sử dụng) đếm số ô của đường đi ngắn nhất giữa 2 ô cho trước trên lưới; nếu không có trả về -1.
 Ví dụ: Theo hình vẽ bên, số ô của đường đi ngắn nhất giữa $[2, 2]$ với $[3, 4]$ là 4; Số ô của đường đi ngắn nhất giữa $[2, 2]$ với $[1, 5]$ là -1.

	1	2	3	4	5
1	0	0	0	0	1
2	1	1	1	1	0
3	1	0	1	1	1
4	1	0	0	1	0
5	1	1	1	1	0

Câu 9 (Câu điểm thưởng, có thể bỏ qua không làm; 1 điểm) – All nearest smaller values

Cho dãy n số nguyên khác nhau đôi một, với phần tử nhỏ nhất ở đầu dãy. Hãy trình bày thuật toán, để với từng phần tử, tìm số ở trước và gần nó nhất trong dãy nhưng có giá trị nhỏ hơn. Đánh giá độ phức tạp của thuật toán? Ví dụ:

Số 0 đầu tiên không có đứng trước. Số nhỏ nhất đứng trước 8 và 4 là 0. Cả ba số đứng trước 12 đều nhỏ hơn nó, nhưng số gần nhất là 4 ...

Input	0	8	4	12	2	10	6	14	1
Output	-	0	0	4	0	2	2	6	0